IN-39
11728
P42

NASA Contractor Report 4358

The Computational Structural Mechanics Testbed Structural Element Processor ES5: STAGS Shell Element

Charles Rankin and Frank Brogan

CONTRACT NAS1-18444
MAY 1991

(NASA-CR-4358) THE COMPUTATIONAL STRUCTURAL MECHANICS TESTBED STRUCTURAL-ELEMENT PROCESSOR ES5: STAGS SHELL ELEMENT (Lockheed Missiles and Space Co.) 42 p CSCL 20K H1/39 N91-22596 Unclas 0011728

NASA Contractor Report 4358

The Computational Structural Mechanics Testbed Structural Element Processor ES5: STAGS Shell Element

Charles Rankin and Frank Brogan
Lockheed Missiles & Space Company, Inc.
Palo Alto, California

Prepared for
Langley Research Center
under Contract NAS1-18444

National Aeronautics and
Space Administration
Office of Management
Scientific and Technical
Information Division

1991

Preface

This report documents the theory behind the CSM Testbed structural finite element processor ES5 for the STAGS shell element. The CSM Testbed is described in reference 1.

This report is intended both for CSM Testbed users, who would like theoretical background on element types before selecting them for an analysis, as well as for element researchers who are attempting to improve existing elements or to develop entirely new formulations.

PRECEDING PAGE BLANK NOT FILMED

PAGE ii INTENTIONALLY BLANK

Contents

PRECEDING PAGE BLANK NOT FILMED

PAGE IV INTENTIONALLY BLANK

List of Figures

List of Tables

Structural Element Processor ES5

1. GENERAL DESCRIPTION

1.1 Purpose

Processor ES5 contains the displacement-based 4-node quadrilateral shell element used as the "workhorse" element in the STAGS code (ref. 2). This C^1 element is intended for modeling thin shells for which transverse-shear deformation is not important. The element employs an incompatible displacement field, which features a high-order (cubic) bending field and a semi-lower-order (linear/cubic) membrane field. For many problems with regular geometry, the STAGS 410 element in ES5, called E410, provides rapid convergence not possible with the same number of freedoms in an element where inter-element compatibility is rigidly enforced. However, it does exhibit distortion sensitivity, especially due to in-plane distortion. Sensitivity to out-of-plane distortion (*i.e.*, warping) is largely eliminated by the generic corotational projection operator built-in to the ES processor shell.

The E410 is a quadrilateral shell element with 3 translational and 3 rotational freedoms per node. This element also has "drilling" rotational stiffness, which proves to be an advantage for problems with intersecting geometries. Hence, no suppression is required for the drilling (*i.e.*, 6th) degree of freedom – even for flat plates – because all 6 nodal freedoms have stiffness.

Arbitrarily large rotations (but only small strains) may be modeled with these elements by employing the standard nonlinear corotational utility available for all ES processors.

1.2 Background

Processor ES5 was developed by Frank Brogan of the Lockheed Palo Alto Research Laboratory. The E410 element was originally developed for the STAGS code over a period of years, and was recently transferred to the CSM Testbed, as processor ES5, under the sponsorship of the NASA CSM Program. The E410 element originated as a plate bending element with the QB12 formulation (ref. 3), with the later addition of the QUAF membrane formulation developed by Bo Almroth and Frank Brogan of the Lockheed Palo Alto Research Laboratory, which is based on an extension of the Ph.D. thesis by Willem (ref. 4).

1.3 Specific Element Types

Processor ES5 contains only one element type, the E410 shell element, which is equivalent to the 410 element within the STAGS code. For quick reference, a summary of element types within processor ES5 is presented in Table 1, and an element fact sheet is provided in Table 2.

PRECEDING PAGE BLANK NOT FILMED

In Table 2 the following definitions apply:

NEN - number of element nodes

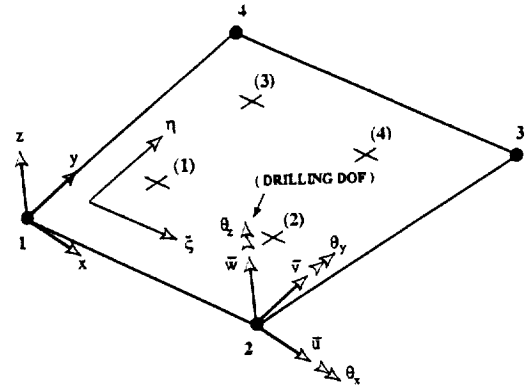
NIP - number of integration points

NSTR - number of stresses

NDOF - number of nodal degrees of freedom

Table 1. Summary of Processor ES5 Element Types

Type	*Description*
E410	4-node quadrilaterial C^1 facet shell element. This is the same element available in the STAGS finite element code (where it is called the 410 element). It is recommended only for experienced users, as it tends to be sensitive to in-plane distortion, and possesses a spurious mode that can occasionally be triggered. However, in the absence of spurious modes and significant mesh distortion, the element can be extremely accurate for thin plate and shell problems.

Table 2. Element ES5/E410 Fact Sheet	
Attribute	*Description*
Element Type	4-Node C^1 Facet Shell Element
Developer	F. A. Brogan, B. O. Almroth, & C. C. Rankin (LPARL)
Topology **NEN=4** **NIP=4** **NSTR=6** **NDOF=6**	 $\mathbf{d}_e^a = \left\{ \begin{array}{c} \overline{\mathbf{u}}_e^a \\ \theta_e^a \end{array} \right\}$ $(a = 1,4)$
Intended Use	Very thin plates and shells
Variational Basis	Assumed displacements (Total Potential Energy)
Geometric Approx.	Bilinear reference surface; normal edges
Displacement Approx.	Linear/cubic u, v; Cubic w (includes drilling DOF)
Strain Approximation $\widetilde{\epsilon}_L = \{\overline{\epsilon}_L, \kappa_L\}^T$	$\overline{\epsilon}_x \sim p_0(\xi)p_2(\eta) \quad \overline{\epsilon}_y \sim p_2(\xi)p_0(\eta) \quad \overline{\epsilon}_{xy} \sim p_2(\xi,\eta)$ $\kappa_x, \kappa_y, \kappa_{xy} \sim p_1(x,y)$ — if elt. edges not at 45° $\kappa_x, \kappa_y, \kappa_{xy} \sim p_1(\xi,\eta)$ — if elt. edges at 45°
Stress Approximation $\widetilde{\sigma}_L = \{\mathbf{n}_L, \mathbf{m}_L\}^T$	Using constitutive relations, e.g., $\widetilde{\sigma}_L(\xi,\eta) = \widetilde{\mathbf{C}}_L(\xi,\eta)\widetilde{\epsilon}_L(\xi,\eta)$
Force Vectors	$\mathbf{f}_e^{int} = \sum_{g=1}^4 w_g\, J(\xi_g,\eta_g)\, \mathbf{B}^T(\xi_g,\eta_g)\, \widetilde{\sigma}(\xi_g,\eta_g)$ $\mathbf{f}_e^{ext} = \sum_{g=1}^4 w_g\, J(\xi_g,\eta_g)\, \widetilde{\mathbf{N}}_D^T(\xi_g,\eta_g)\, \widetilde{\mathbf{f}}^{ext}(\xi_g,\eta_g)$
Stiffness Matrices	$\mathbf{K}_e^{matl} = \sum_{g=1}^4 w_g\, J(\xi_g,\eta_g) \mathbf{B}^T(\xi_g,\eta_g)\, \widetilde{\mathbf{C}}(\xi_g,\eta_g)\, \mathbf{B}(\xi_g,\eta_g)$ $\mathbf{K}_e^{geom} = \sum_{g=1}^4 w_g\, J(\xi_g,\eta_g) \mathbf{G}^{iT}(\xi_g,\eta_g) \mathbf{S}(\xi_g,\eta_g) \mathbf{G}^i(\xi_g,\eta_g)$
Mass Matrices	$\mathbf{M}_e^C = \sum_{g=1}^4 w_g\, J(\xi_g,\eta_g)\, \widetilde{\mathbf{N}}_D^T(\xi_g,\eta_g)\, \mathcal{I}\, \widetilde{\mathbf{N}}_D^T(\xi_g,\eta_g)$ $\mathbf{M}_e^D = diag\{(\overline{m}_1\mathbf{I}_3, \widehat{m}_1\mathbf{I}_3), \ldots, (\overline{m}_4\mathbf{I}_3, \widehat{m}_4\mathbf{I}_3)\}$
Nonlinearity	Lagrange strains and/or corotation
Pathologies	Distortion sensitive; potential spurious modes
Recommended Use	General-purpose application by experienced users

2. ELEMENT FORMULATION

2.1 Summary

The E410 element is a C^1 (slope-continuous) element based on the Kirchhoff-Love shell hypothesis (normals stay normal, no transverse-shear deformation) with bending energy dependent on a single transverse displacement field. The membrane energy relies on a bilinear interpolation of the in-plane nodal translations and a cubic interpolation of the normal nodal rotations. The bending energy relies on a cubic interpolation of the transverse nodal displacements and bending rotations. The result is an element with membrane strains that are constant along the direction of a given strain component, but vary quadratically in the lateral direction; and with bending strains that vary linearly in both directions. Note that the E410 element is a "flat", facet-type element. That is, all geometry is based on a projection of the actual geometry onto a best-fit plane. This element is therefore sensitive to warping, unless the corotational projection option is invoked when employing CSM Testbed processor ES5. The combination of high-order bending formulation, drilling rotational stiffness, and the generic corotational facility allows for accurate and efficient computation of arbitrarily large rotation, moderate strain response — provided that in-plane element distortion is only mild (*i.e.*, the projected element shape should be kept near-rectangular).

2.2 Variational Basis

The E410 element can be derived by starting with the principle of minimum total potential energy, wherein the displacements are the only independently approximated field.

2.2.1 Continuum Equations

The principle of minimum total potential energy states that:

$$\delta \Pi_T(\mathbf{u}) \;=\; 0 \tag{1}$$

where, for linear elastic analysis, Π_T is the total potential energy functional:

$$\Pi_T(\mathbf{u}) \;=\; \frac{1}{2}\int_V \epsilon(\mathbf{u})^T\,\mathbf{C}\,\epsilon(\mathbf{u})\,dV \;-\; \Big(\int_V \mathbf{u}^T\,\mathbf{f}^b\,dV \;+\; \int_S \mathbf{u}^T\,\mathbf{f}^s\,dS\Big) \tag{2}$$

in which $\mathbf{u} = \mathbf{u}(\mathbf{x})$ is the displacement vector, $\mathbf{x}$ is the position vector, $\mathbf{f}^b$ and $\mathbf{f}^s$ are body and surface force vectors, $\mathbf{C} \;=\; \mathbf{C}(\mathbf{x})$ is the constitutive matrix, and the strain operator, $\epsilon(\mathbf{u})$, is defined for linear analysis by:

$$\epsilon(\mathbf{u}) \;=\; \frac{1}{2}\big(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\big) \tag{3}$$

which in vector notation and Cartesian components is just:

$$\epsilon \;=\; \{\,\epsilon_{11} \quad \epsilon_{22} \quad 2\epsilon_{12} \quad 2\epsilon_{31} \quad 2\epsilon_{32} \quad \epsilon_{33}\,\}^T \tag{4}$$

2.2.2 Shell Assumptions

The geometry of a general shell is illustrated in Figure 1; the E410 element formulation, however, is based upon the restriction of the shell to a *flat* plate, which may be thought of as a facet model of the shell. The following assumptions are introduced in the continuum variational equations to obtain corresponding "shell" variational equations — and hence reduce the above volume integrals to surface integrals:

1) Shell normals remain straight *and* normal (zero transverse-shear strain).

2) The shell normal stress can be neglected.

The displacement field depends on the position of the reference surface plus the normal distance z times the derivatives of the transverse displacement, *i.e.*,

$$\boxed{\begin{aligned} u(x,y,z) &= \overline{u}(x,y) - z\,\overline{w}_{,x}(x,y) \\ v(x,y,z) &= \overline{v}(x,y) - z\,\overline{w}_{,y}(x,y) \\ w(x,y,z) &= \overline{w}(x,y) \end{aligned}} \tag{5}$$

where x and y are the in-plane local-Cartesian coordinates along the *reference surface* (defined by $z = 0$); u, v, w are the displacements in the x, y, z directions of a point located a distance z off (*i.e.*, normal to) the reference surface; $\overline{u}, \overline{v}, \overline{w}$ are the x, y, z-directed displacement components of a corresponding point *on* the reference surface; and $\overline{w}_{,x}$ and $\overline{w}_{,y}$ are the derivative of $\overline{w}$ with respect to x and y, respectively.

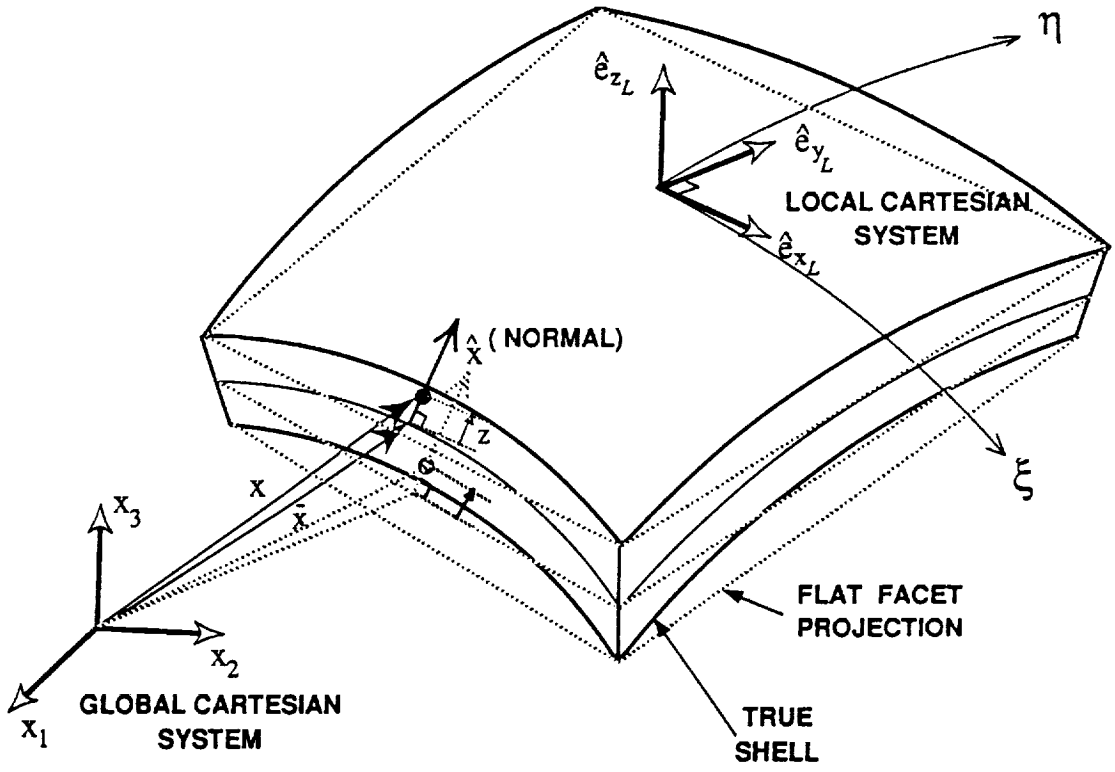


Figure 1. Shell Geometry Basis of ES5/E410 Element

Note that $\overline{w}_{,x}$ and $\overline{w}_{,y}$ are associated with the deformed slopes of the reference surface, which are related to engineering rotations as follows:

$$\begin{aligned} \theta_x(x,y) &= \overline{w}_{,y}(x,y) \\ \theta_y(x,y) &= -\overline{w}_{,x}(x,y) \end{aligned} \tag{6}$$

These engineering rotations will be employed, subsequently, as element nodal rotational freedoms, with an additional *drilling* rotational freedom defined within the interior as follows:

$$\theta_z(x,y) = \frac{1}{2}(\overline{u}_{,y} - \overline{v}_{,x}) \tag{7}$$

It is convenient to re-express equation (5) using the following vector notation:

$$\boxed{\mathbf{u}(x,y,z) = \overline{\mathbf{u}}(x,y) + z\,\mathbf{u}'(x,y)} \tag{8}$$

where

$$\begin{aligned} \mathbf{u} &= \{\, u \quad v \quad w \,\}^T \\ \overline{\mathbf{u}} &= \{\, \overline{u} \quad \overline{v} \quad \overline{w} \,\}^T \\ \mathbf{u}' &= \{\, -\overline{w}_{,x} \quad -\overline{w}_{,y} \quad 0 \,\}^T \end{aligned} \tag{9}$$

Substituting equation (8) into equation (2) yields the following "shell" total potential energy functional:

$$\boxed{\widetilde{\Pi}_T(\overline{\mathbf{u}}) = \frac{1}{2}\int_S \widetilde{\epsilon}(\overline{\mathbf{u}})^T\,\widetilde{\mathbf{C}}\,\widetilde{\epsilon}(\overline{\mathbf{u}})\,dS - \left(\int_S \widetilde{\mathbf{u}}^T(\widetilde{\mathbf{f}}^b + \widetilde{\mathbf{f}}^s)\,dS + \int_L \widetilde{\mathbf{u}}^T\widetilde{\mathbf{f}}^\ell\,dL\right)} \tag{10}$$

where the superposed tildes represent shell resultant quantities, defined as follows:

$$\widetilde{\mathbf{u}} = \{\, \overline{\mathbf{u}} \quad \mathbf{u}' \,\}^T = \text{shell displacement vector} \tag{11a}$$

$$\widehat{\epsilon} = \mathbf{Z}\,\widetilde{\epsilon} = \text{reduced strain vector} \tag{11b}$$

$$\widetilde{\mathbf{C}} = \int_z \mathbf{Z}^T\,\widehat{\mathbf{C}}\,\mathbf{Z}\,dz = \text{resultant constitutive matrix} \tag{11c}$$

$$\widetilde{\sigma} = \int_z \mathbf{Z}^T\,\widehat{\sigma}\,dz = \text{stress resultant vector} \tag{11d}$$

$$\widetilde{\mathbf{f}}^b = \{\, \textstyle\int_z \mathbf{f}^b\,dz\,, \quad \int_z \mathbf{f}^b\,z\,dz \,\}^T = \text{shell body force} \tag{11e}$$

$$\widetilde{\mathbf{f}}^s = \{\, \textstyle\int_z \mathbf{f}^s\,dz\,, \quad \int_z \mathbf{f}^s\,z\,dz \,\}^T = \text{shell surface force} \tag{11f}$$

$$\widetilde{\mathbf{f}}^\ell = \{\, \textstyle\int_z \mathbf{f}^s\,dz\,, \quad \int_z \mathbf{f}^s\,z\,dz \,\}^T = \text{shell line load} \tag{11g}$$

in which

$$\mathbf{Z}(z) = [\mathbf{I}_3 \quad z\mathbf{I}_3] \tag{12}$$

Note that the hats above $\widehat{\epsilon}$, $\widehat{\sigma}$ and $\widehat{\mathrm{C}}$ denote enforcement of both the zero transverse-shear strain assumption, and the zero normal stress assumption, *i.e.*,

$$\epsilon_{xz} = \epsilon_{zy} = \sigma_{zz} = 0 \tag{13}$$

so that the dimension of these arrays has been reduced from 6 to 3, *i.e.*,

$$\widehat{\epsilon} = \{\, \epsilon_{xx} \quad \epsilon_{yy} \quad 2\epsilon_{xy} \,\}^T \qquad \text{and} \qquad \widehat{\sigma} = \{\, \sigma_{xx} \quad \sigma_{yy} \quad \sigma_{xy} \,\}^T \tag{14}$$

and

$$\widehat{\mathrm{C}} = \begin{array}{c} \\ \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{array} \begin{array}{c} \begin{array}{ccc} \epsilon_{xx} & \epsilon_{yy} & \epsilon_{xy} \end{array} \\ \begin{pmatrix} \widehat{C}_{11} & \widehat{C}_{12} & \widehat{C}_{13} \\ \widehat{C}_{12} & \widehat{C}_{22} & \widehat{C}_{23} \\ \widehat{C}_{13} & \widehat{C}_{23} & \widehat{C}_{33} \end{pmatrix} \end{array} \tag{15}$$

where

$$\widehat{C}_{ij} = C_{ij} - \frac{C_{i6}\, C_{6j}}{C^L_{66}} \qquad (i,j = 1:3) \tag{16}$$

The normal strain is then always recoverable using:

$$\epsilon_{zz} = \sum_{j=1}^{3} \frac{C_{6j}\, \epsilon_j}{C_{66}} \tag{17}$$

where subscripts $i,j = 1,2,3,6$ denote coefficients relating stress/strain components xx, yy, xy and zz, respectively.

Shell Strain Measures

The shell resultant strain measures, $\widetilde{\epsilon}$, are defined in terms of displacements expressed in the local-Cartesian basis as follows:

$$\widetilde{\epsilon} = \left\{ \begin{matrix} \overline{\epsilon} \\ \kappa \end{matrix} \right\} = \left\{ \begin{matrix} \text{membrane strains} \\ \text{bending strains} \end{matrix} \right\} \tag{18}$$

where

$$\boxed{\begin{aligned} \overline{\epsilon} &= \left\{ \begin{matrix} \overline{\epsilon}_x \\ \overline{\epsilon}_y \\ \overline{\epsilon}_{xy} \end{matrix} \right\} = \left\{ \begin{matrix} \dfrac{\partial \overline{u}}{\partial \overline{x}} \\ \dfrac{\partial \overline{v}}{\partial \overline{y}} \\ \dfrac{\partial \overline{u}}{\partial \overline{y}} + \dfrac{\partial \overline{v}}{\partial \overline{x}} \end{matrix} \right\} \\ \kappa &= \left\{ \begin{matrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{matrix} \right\} = \left\{ \begin{matrix} -\dfrac{\partial^2 \overline{w}}{\partial \overline{x} \partial \overline{x}} \\ -\dfrac{\partial^2 \overline{w}}{\partial \overline{y} \partial \overline{y}} \\ -2\dfrac{\partial^2 \overline{w}}{\partial \overline{x} \partial \overline{y}} \end{matrix} \right\} \end{aligned}} \tag{19}$$

Note that the continuum strains, excluding transverse shear and normal components, can be computed from the above resultant strains using equation (11b), *i.e.*,

$$\widehat{\epsilon} = \overline{\epsilon} + z\,\kappa \tag{20}$$

Shell Stress Resultants

The shell stress resultants are defined, from equation (11d), as follows:

$$\widetilde{\sigma} = \left\{ \begin{matrix} \mathbf{n} \\ \mathbf{m} \end{matrix} \right\} = \left\{ \begin{matrix} \text{membrane stress resultants (force/length)} \\ \text{bending stress resultants (moment/length)} \end{matrix} \right\} \tag{21}$$

where

$$\boxed{\begin{aligned} \mathbf{n} &= \int_z \widehat{\sigma}\, dz = \left\{ \begin{matrix} n_x \\ n_y \end{matrix} \right\} = \int_z \left\{ \begin{matrix} \sigma_{xx} \\ \sigma_{yy} \end{matrix} \right\} dz \\ \mathbf{m} &= \int_z \widehat{\sigma}\, z\, dz = \left\{ \begin{matrix} m_x \\ m_y \end{matrix} \right\} = \int_z \left\{ \begin{matrix} \sigma_{xx} \\ \sigma_{yy} \end{matrix} \right\} z\, dz \end{aligned}} \tag{22}$$

2.3 Discrete Equations

The finite element shell equations are obtained from equation (10) by introducing approximations for the geometry (reference surface coordinates) and displacement field within each element of the form:

$$\mathbf{x}(\xi,\eta) = \mathbf{N}^G(\xi,\eta)\,\mathbf{x}^e \tag{23a}$$

$$\overline{\mathbf{u}}(\xi,\eta) = \mathbf{N}^D(\xi,\eta)\,\mathbf{d}^e \tag{23b}$$

where

$$\mathbf{x}^e = \begin{Bmatrix} \mathbf{x}_1^e \\ \mathbf{x}_2^e \\ \mathbf{x}_3^e \\ \mathbf{x}_4^e \end{Bmatrix} \quad \text{and} \quad \mathbf{d}^e = \begin{Bmatrix} \mathbf{d}_1^e \\ \mathbf{d}_2^e \\ \mathbf{d}_3^e \\ \mathbf{d}_4^e \end{Bmatrix} \tag{24}$$

are the expanded element nodal coordinate and nodal displacement vectors, respectively, and ξ,η are natural surface coordinates ranging between -1 and +1 within the element. With the above discrete approximations (defined in detail in the following sections), the resultant strains can be computed using:

$$\begin{aligned} \widetilde{\epsilon}(\xi,\eta) &= \widetilde{\epsilon}\Big(\mathbf{N}^D(\xi,\eta)\,\mathbf{d}^e\Big) \\ &= \mathbf{B}(\xi,\eta)\,\mathbf{d}^e \end{aligned} \tag{25}$$

where $\mathbf{B}$ is the element strain-displacement matrix, defined by substituting the element displacement approximations (eq. (23b)) into the strain-displacement relations (eqs. (18)-(19)).

The discrete form of the variational functional (eq. (10)) thus becomes:

$$\Pi_T = \sum_{e=1}^{Nel} \Pi_T^e \tag{26}$$

where the script e denotes an individual element, Nel is the total number of elements, and the element total potential energy may be expressed as:

$$\boxed{\Pi_T^e = \frac{1}{2}\mathbf{d}_e^T\,\mathbf{K}_e^{matl}\,\mathbf{d}_e - \mathbf{d}_e^T\,\mathbf{f}_e^{ext}} \tag{27}$$

In equation (27) $\mathbf{K}_e^{matl}$ is the element *material* (or linear) stiffness matrix, and $\mathbf{f}_e^{ext}$ is the element external force vector, defined as follows:

$$\begin{aligned} \mathbf{K}_e^{matl} &= \mathbf{T}_{EC}^T \Big(\int_S \overline{\mathbf{B}}^T\,\widetilde{\mathbf{C}}\,\overline{\mathbf{B}}\,dS \Big)\,\mathbf{T}_{EC} \\ \mathbf{f}_e^{ext} &= \mathbf{T}_{EC}^T \Big(\underbrace{\int_S \mathbf{N}_D^T\,\widetilde{\mathbf{f}}^b\,dS}_{\mathbf{f}_e^{body}} + \underbrace{\int_S \mathbf{N}_D^T\,\widetilde{\mathbf{f}}^s\,dS}_{\mathbf{f}_e^{surf}} + \underbrace{\int_L \mathbf{N}_D^T\,\widetilde{\mathbf{f}}\,dL}_{\mathbf{f}_e^{line}} \Big) \end{aligned} \tag{28}$$

where $\mathbf{T}_{EC}$ is the block-diagonal transformation matrix relating the computational basis at each element node to the element local-Cartesian basis. Specific equation systems emanating from equations (26)-(27), and their generalizations, are described in the following sections.

2.3.1 Linear Static Equations

The discrete equations for linear statics are obtained by taking the first variation of equation (26) and setting it equal to zero, *i.e.*,

$$\mathbf{K}^{matl}\mathbf{d} = \mathbf{f}^{ext} \tag{29}$$

where $\mathbf{K}^{matl}$ and $\mathbf{f}^{ext}$ are the assembled versions of the element material stiffness and external force vector (eq. (28)), and $\mathbf{d}$ is the system displacement vector, composed of the union of all element nodal displacement vectors.

2.3.2 Linear Dynamic Equations

For linear dynamics, an inertial term is added to the left-hand side of equation (29) – using Hamilton's principle – resulting in:

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{K}^{matl}\mathbf{d} = \mathbf{f}^{ext} \tag{30}$$

where $\mathbf{M}$ is the structure mass matrix, assembled from the element mass matrices, $\mathbf{M}_e$, which are defined in Section 2.11.

2.3.3 Linear Eigenproblems

For linear vibration analysis, the right-hand-side of equation (30) becomes zero and we have the eigenproblem:

$$\left(\mathbf{K}^{matl} + \lambda\mathbf{M}\right)\mathbf{d}_\lambda = \mathbf{0} \tag{31}$$

where the eigenvalue, λ, is the natural frequency squared, and $\mathbf{d}_\lambda$ is the corresponding vibration mode vector.

For linear stability, or buckling analysis, $\mathbf{M}$ is replaced with the geometric stiffness matrix:

$$\left(\mathbf{K}^{matl} + \lambda\mathbf{K}^{geom}(\sigma_0)\right)\mathbf{d}_\lambda = \mathbf{0} \tag{32}$$

where the eigenvalue, λ, is the buckling pre-stress load multiplier, (σ_0) is the buckling pre-stress field, $\mathbf{d}_\lambda$ is the corresponding buckling mode, and $\mathbf{K}^{geom}$ is the geometric stiffness matrix, defined in Section 2.10.

2.4 Element Topology

The topology of the E410 shell element in processor ES5 is shown in Figure 2.

Each node possesses the 6 displacement degrees of freedom:

$$\mathbf{d}_a = \left\{ \begin{array}{c} \overline{\mathbf{u}}_a \\ \boldsymbol{\theta}_a \end{array} \right\} \tag{33}$$

where:

$$\overline{\mathbf{u}}_a = \left\{ \begin{array}{c} \overline{u}_a \\ \overline{v}_a \\ \overline{w}_a \end{array} \right\} \qquad \text{and} \qquad \boldsymbol{\theta}_a = \left\{ \begin{array}{c} \theta_{xa} \\ \theta_{ya} \\ \theta_{za} \end{array} \right\} \tag{34}$$

are the translation and rotation components at element node a, expressed in the fixed element local x, y, z coordinate system (in which z is the shell-facet normal direction).

Each element also stores the 6 resultant stress and/or strain components defined in equations (19) and (21) at each of the 2 by 2 Gauss integration points illustrated in Figure 2.

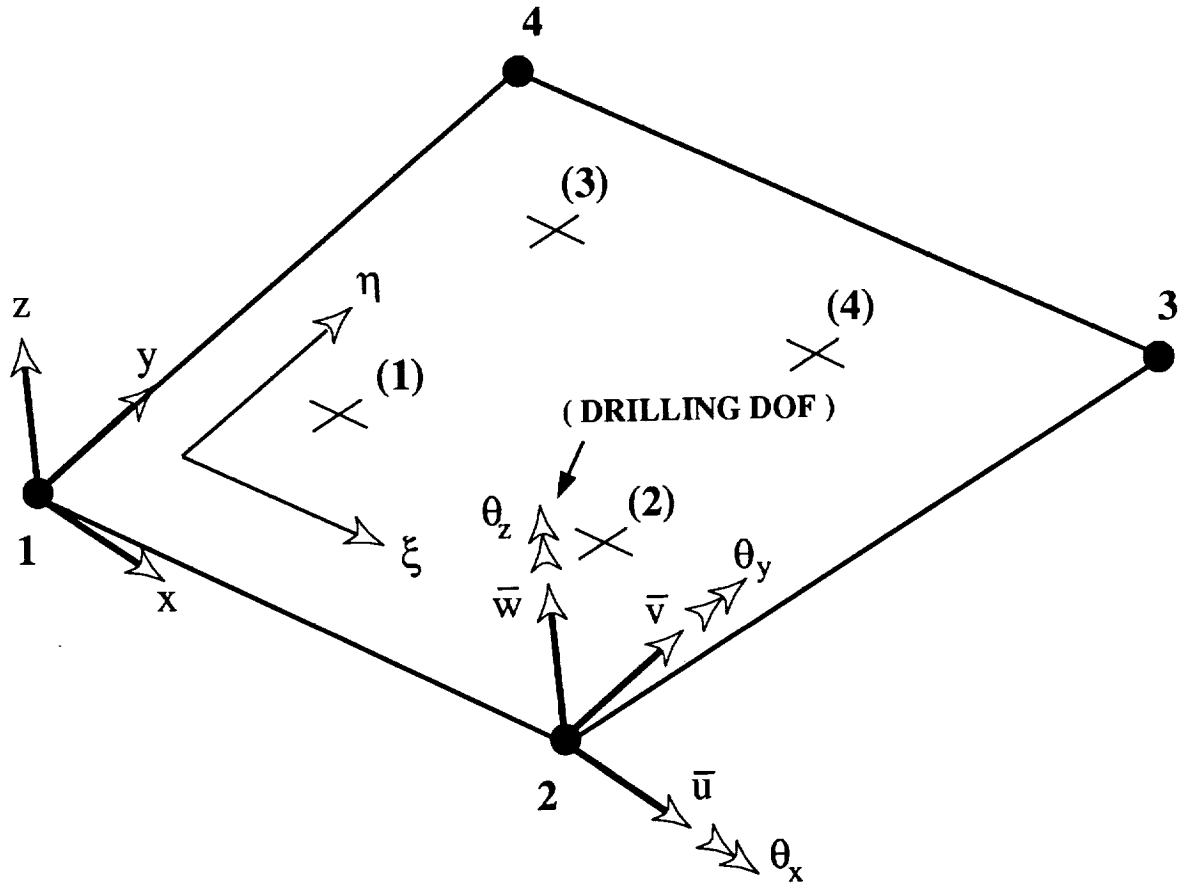


Figure 2. Topology of ES5/E410 Shell Element

2.5 Geometric Approximations

The element geometry, embodied in equation (23a), is approximated within the element by interpolating reference surface coordinates from corresponding *projected* (flat-facet) nodal quantities. To accomplish a best-fit, a plane is chosen according to the prescription described in Chapter 3 of reference 5 such that nodes 1 and 3 lie in the plane, and nodes 2 and 4 lie an equal distance away on the same side (see Figure 2). Actual nodal positions are projected onto the plane and become the $\mathbf{x}_a$ vectors used in equation (23a). Interior reference surface coordinates are obtained using bilinear Lagrange interpolation of the projected nodal coordinate vectors, *i.e.*,

$$\mathbf{x}(\xi,\eta) \;=\; \sum_{a=1}^{4} N_a(\xi,\eta)\,\mathbf{x}_a \tag{35}$$

in which ξ,η are the natural coordinates of the quadrilateral element (see Figure 2), and $N_a(\xi,\eta)$ is the Lagrange interpolating polynomial – or shape function – associated with element node a.

The above expression may be recast in the matrix notation of equation (23a) as follows:

$$\mathbf{x}(\xi,\eta) \;=\; \mathbf{N}^G(\xi,\eta)\,\mathbf{x}^e \tag{36}$$

where

$$\mathbf{x}(\xi,\eta) \;=\; \left\{ \begin{array}{c} x(\xi,\eta) \\ y(\xi,\eta) \end{array} \right\} \tag{37}$$

$$\mathbf{x}^e \;=\; \left\{ \begin{array}{c} \mathbf{x}_1^e \\ \mathbf{x}_2^e \\ \mathbf{x}_3^e \\ \mathbf{x}_4^e \end{array} \right\} \qquad \mathbf{x}_a^e \;=\; \left\{ \begin{array}{c} x_a \\ y_a \end{array} \right\} \tag{38}$$

and

$$\boxed{\mathbf{N}^G(\xi,\eta) \;=\; [\,N_1(\xi,\eta)\mathbf{I}_2 \quad N_2(\xi,\eta)\mathbf{I}_2 \quad N_3(\xi,\eta)\mathbf{I}_2 \quad N_4(\xi,\eta)\mathbf{I}_2\,]} \tag{39}$$

2.6 Displacement Approximations

The E410 element displacement field may be divided into two groups, as follows:

1) The *transverse* – $\overline{w}$ – field, which is interpolated from nodal freedoms $\{w_a, \theta_{xa}, \theta_{ya}\}$. From this field the bending curvatures $\kappa_x, \kappa_y, \kappa_{xy}$ are computed for the complete description of the bending response. For this purpose, the bending interpolations suggested by Melosh (ref. 3) are used, with a special modification made to avoid singularities (when adjacent element sides form an angle of 45 degrees).

2) The *in-plane* – $\overline{u}, \overline{v}$ – field, which is interpolated from nodal freedoms $\{\overline{u}_a, \overline{v}_a, \theta_{za}\}$. From this field the membrane strains, $\overline{\epsilon}_x, \overline{\epsilon}_y, \overline{\epsilon}_{xy}$ are computed for the complete description of the membrane response. For this purpose, a modified version of the membrane interpolations suggested by Willem (ref. 4) is used, which employs the drilling freedoms θ_{za}, in such a way that compatibility between in-plane and transverse displacements is achieved when two elements intersect at 90 degrees (*i.e.*, box-like structures).

2.6.1 Bending Displacement Field

The approximation of the E410 bending, or transverse, displacement field may be expressed as follows:

$$\overline{\mathbf{u}}^B(x,y) \;=\; \mathbf{N}^B(x,y)\;\mathbf{d}^B \tag{40}$$

where

$$\overline{\mathbf{u}}^B(x,y) = \{\,\overline{w}(x,y)\,\} \tag{41}$$

and

$$\mathbf{d}^B = \begin{Bmatrix} \mathbf{d}_1^B \\ \mathbf{d}_2^B \\ \mathbf{d}_3^B \\ \mathbf{d}_4^B \end{Bmatrix} \qquad \mathbf{d}_a^B = \begin{Bmatrix} \overline{w}_a \\ \theta_{xa} \\ \theta_{ya} \end{Bmatrix} \tag{42}$$

The bending displacement interpolation matrix, $\mathbf{N}^B$, may be partitioned into nodal blocks:

$$\mathbf{N}^B \;=\; [\mathbf{N}_1^B \quad \mathbf{N}_2^B \quad \mathbf{N}_3^B \quad \mathbf{N}_4^B\,] \tag{43}$$

with

$$\boxed{\mathbf{N}_a^B(\xi,\eta) \;=\; \begin{matrix} \\ \overline{w} \end{matrix} \begin{matrix} \overline{w}_a & \theta_{xa} & \theta_{ya} \\ \begin{pmatrix} N^w_{w_a} & N^w_{\theta_{xa}} & N^w_{\theta_{ya}} \end{pmatrix} \end{matrix}} \tag{44}$$

The bending interpolation functions, $N^w_{ia}(x,y)$, are constructed from a general polynomial expansion complete through third order, with the addition of the fourth order terms x^3y and xy^3, as follows:

$$\boxed{N^w_{ia}(x,y) = c^{00}_{ia} + c^{10}_{ia}x + c^{01}_{ia}y + c^{11}_{ia}xy + \ldots + c^{13}_{ia}xy^3} \tag{45}$$

where the 12 unknown coefficients c^{kl}_{ia} depend only on the element geometry. The superscripts k,l label the x and y polynomial powers, respectively, while the subscript a labels the node, and the subscript i labels freedoms $\overline{w}, \theta_x, \theta_y$ respectively. Thus, there are 12 sets of 12 coefficients, or one interpolation function for each nodal bending freedom.

For $N^w_{ia}(x,y)$ to describe a true interpolation function, its value at the nodes must be zero for all freedoms except ia, where it must equal 1. This provides each of the linear equations for the 12 unknown c^{kl} coefficients. The rotational freedoms are matched to the derivatives $N^w_{ia,x}$ or $N^w_{ia,y}$ at the nodes, as required by linear theory, *i.e.*,

$$\begin{aligned} \theta_{xa} &= \overline{w}_{a,y} \\ \theta_{ya} &= -\overline{w}_{a,x} \end{aligned} \tag{46}$$

The above set of 12 linear equations is solved only once per element, as soon as the initial element geometry is known (there is actually one factorization followed by solution for 12 right-hand sides). It can be shown that the resulting displacement field possesses the correct rigid-body modes, and, for rectangular and parallelogram shapes, correctly describes a general state of constant curvature (see, *e.g.*, Melosh (ref. 3)).

Note that once $\overline{\mathbf{u}}^B(x,y)$ is known from equation (40), the corresponding function of (ξ,η) can be obtained by making the simple substitution:

$$\overline{\mathbf{u}}^B(x,y) \;=\; \overline{\mathbf{u}}^B(x(\xi,\eta), y(\xi,\eta)) \;=\; \overline{\mathbf{U}}^B(\xi,\eta) \tag{47}$$

where $\mathbf{x}(\xi,\eta)$ is obtained using equation (35). This change of variables is useful when performing numerical integration over the element domain, since numerical quadrature rules are typically expressed in terms of natural coordinates.

Special Treatment of Bending Displacement Pathologies

The presence of the xy^3 and x^3y terms in the approximation of the bending displacement field introduces the possibility of a *mechanism* (*i.e.*, spurious mode) in certain instances, particularly where two free edges intersect. In the STAGS code, an optional penalty term is added to suppress this behavior; however, in the CSM Testbed the penalty option has not been implemented.*

Another pathology – in very rare cases – is that the system of equations for the coefficients in equation (45) becomes *singular* when two element edges intersect at exactly 45 degrees. This makes it impossible to construct the bending interpolation functions. A satisfactory way around this is to replace the polynomial expansion given in equation (45) in terms of x, y, with an identical expansion in terms of ξ, η – the natural coordinates. Since the natural coordinates refer to a "parent" element that is always square, the singularity never arises. This leads to new interpolation functions, $N^w_{ia}(\xi, \eta)$, which have been given explicitly by Melosh (ref. 3) – as shown in Table 3 – so that solution of a 12×12 equation system is avoided altogether. However, the trade-off is that these interpolation functions must be doubly differentiated with respect to x, y to obtain the bending curvatures, which requires a somewhat complicated Jacobian transformation (see Section 2.7.2 for details).

Table 3. Explicit Natural-Coordinate Bending Shape Functions

$N^w_{w_a}(\xi,\eta)$	$\frac{1}{8}\left[(\xi_a\xi+1)(\eta_a\eta+1)(2+\xi_a\xi+\eta_a\eta-\xi^2-\eta^2)\right]$
$N^w_{w,\xi_a}(\xi,\eta)$	$\frac{1}{8}\left[\xi_a(\xi_a\xi+1)^2(\xi_a\xi-1)(\eta_a\eta+1)\right]$
$N^w_{w,\eta_a}(\xi,\eta)$	$\frac{1}{8}\left[\eta_a(\eta_a\eta+1)^2(\eta_a\eta-1)(\xi_a\xi+1)\right]$

* Note that this mechanism is usually alleviated in nonlinear analysis by the corotational projection operator.

2.6.2 Membrane Displacement Field

The approximation of the E410 membrane, or in-plane, displacement field may be expressed as follows:

$$\overline{\mathbf{u}}^M(\xi,\eta) \;=\; \mathbf{N}^M(\xi,\eta)\;\mathbf{d}^M \tag{48}$$

where

$$\overline{\mathbf{u}}^M(x,y) = \left\{ \begin{array}{c} \overline{u}(\xi,\eta) \\ \\ \overline{v}(\xi,\eta) \end{array} \right\} \tag{49}$$

and

$$\mathbf{d}^M = \left\{ \begin{array}{c} \mathbf{d}_1^M \\ \\ \mathbf{d}_2^M \\ \\ \mathbf{d}_3^M \\ \\ \mathbf{d}_4^M \end{array} \right\} \qquad \text{with} \qquad \mathbf{d}_a^M = \left\{ \begin{array}{c} \overline{u}_a \\ \\ \overline{v}_a \\ \\ \theta_{za} \end{array} \right\} \tag{50}$$

The membrane displacement interpolation matrix, $\mathbf{N}^M$, may be partitioned into nodal blocks as follows:

$$\mathbf{N}^M \;=\; [\,\mathbf{N}_1^M \quad \mathbf{N}_2^M \quad \mathbf{N}_3^M \quad \mathbf{N}_4^M\,] \tag{51}$$

where

$$\mathbf{N}_a^M \;=\; \begin{array}{c} \\ \overline{u} \\ \overline{v} \end{array} \begin{array}{c} \begin{array}{ccc} \overline{u}_a & \overline{v}_a & \theta_{za} \end{array} \\ \left(\begin{array}{ccc} N^u_{u_a} & N^u_{v_a} & N^u_{\theta_{za}} \\ \\ N^v_{u_a} & N^v_{v_a} & N^v_{\theta_{za}} \end{array} \right) \end{array} \tag{52}$$

To obtain the membrane interpolation functions, N^u_{ia}, and N^v_{ia}, where $i = 1,2,3$ corresponds to u, v, θ_z respectively, we *begin* with the basic bilinear displacement field:

$$\begin{aligned} \overline{u}(\xi,\eta) \;&=\; \sum_{i=1}^{4} N_a(\xi,\eta)\,\overline{u}_a \\ \overline{v}(\xi,\eta) \;&=\; \sum_{i=1}^{4} N_a(\xi,\eta)\,\overline{v}_a \end{aligned} \tag{53}$$

where the interpolation functions $N_a(\xi,\eta)$ are the standard bilinear Lagrange functions:

$$N_a(\xi,\eta) \;=\; \frac{1}{4}(1+\xi/\xi_a)(1+\eta/\eta_a) \tag{54}$$

Here, ξ,η are the natural coordinates of an arbitrary point in the element interior, and ξ_a, η_a are the values of ξ,η at element node a, which are given in Table 4.

Table 4. Surface Coordinates at Nodes		
Node (a)	ξ_a	η_a
1	-1	-1
2	1	-1
3	1	1
4	-1	1

It can be seen that the above bilinear interpolation would properly interpolate the $(\overline{u}, \overline{v})$ field, were it not for the requirement that comes from incorporating the θ_{za} drilling freedoms. The drilling freedom can be viewed as a simultaneous (rigid) rotation of any two element edges that intersect at a node, as shown in Figure 3.

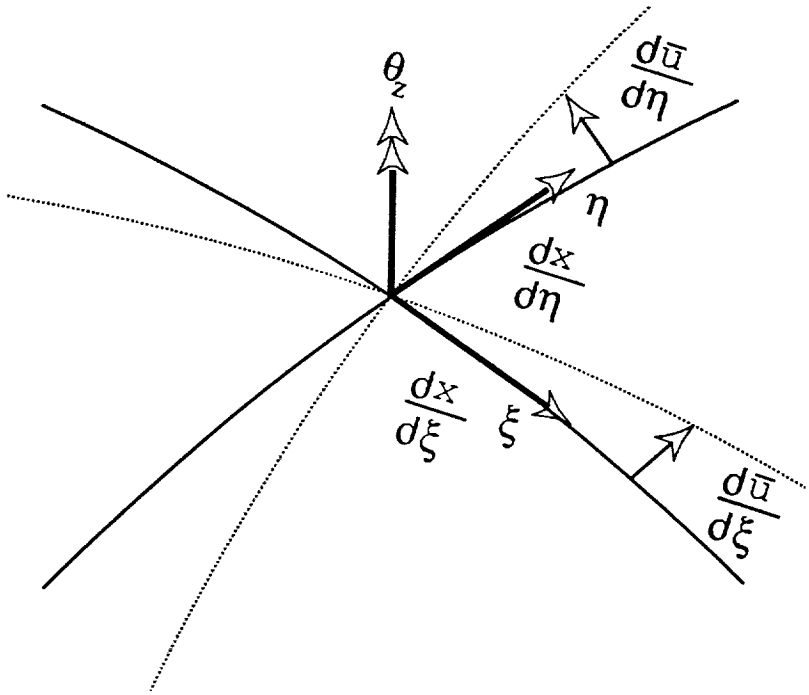


Figure 3. Interpretation of ES5/E410 Drilling Freedoms

For small rotations, a suitable definition of an incremental rotation vector, $\Delta\boldsymbol{\theta}$, is:

$$\Delta\boldsymbol{\theta} = \frac{\mathbf{r} \times \Delta\mathbf{r}}{|\mathbf{r}|^2} \tag{55}$$

where $\mathbf{r}$ is an arbitrary position vector, and $\Delta\mathbf{r}$ is the incremental displacement of that vector under the rigid-body rotation $\Delta\boldsymbol{\theta}$. This definition is consistent with the well-known relation from classical dynamics:*

$$\Delta\mathbf{r} = \Delta\boldsymbol{\theta} \times \mathbf{r} \tag{56}$$

which, when substituted into the right-hand side of equation (55), verifies that expression.

For *flat* elements like E410, the z component of equation (55), with $\theta_z \approx \Delta\theta_z$, translates into:

$$\theta_z = \frac{x_{,\xi}\overline{v}_{,\xi} - y_{,\xi}\overline{u}_{,\xi}}{x_{,\xi}^2 + y_{,\xi}^2} \tag{57}$$

for the rotation of an element edge corresponding to a constant η line; and into:

$$\theta_z = \frac{x_{,\eta}\overline{v}_{,\eta} - y_{,\eta}\overline{u}_{,\eta}}{x_{,\eta}^2 + y_{,\eta}^2} \tag{58}$$

for the rotation of an element edge corresponding to a constant ξ line (see Figure 3). For any two intersecting element edges, *both* of the above equations must be simultaneously satisfied, yielding a total of 8 conditions (2 for each node). We therefore require 8 additional independent interpolation functions, all of which contribute zero $\overline{u}$ and $\overline{v}$ at nodes (to satisfy the basic interpolation condition), but possess nonzero derivatives at nodes for use in equations (57)-(58). The following set of functions serves this purpose:

$$\begin{aligned} \overline{N}_a^u(\xi,\eta) &= \frac{1}{16}(2 + 3\xi\xi_a - \xi^3\xi_a)(\eta^3 + \eta^2\eta_a - \eta - \eta_a) \\ \overline{N}_a^v(\xi,\eta) &= \frac{1}{16}(\xi^3 + \xi^2\xi_a - \xi - \xi_a)(2 + 3\eta\eta_a - \eta^3\eta_a) \end{aligned} \tag{59}$$

for interpolating $\overline{u}$ and $\overline{v}$ displacement fields, respectively. Note that unlike standard shape functions, the edge rotation conditions equations (57)-(58) couple the $\overline{u}$ and $\overline{v}$ fields together, so that $\overline{v}$ displacements in the interior of the element arise from $\overline{u}_a$ nodal displacements, and *vice versa*.

---

* The notation used in classical dynamics is typically $\mathbf{v} = \boldsymbol{\omega} \times \mathbf{r}$, where $\mathbf{v}$ is the translational velocity (corr. to $\Delta\mathbf{r}$) and $\boldsymbol{\omega}$ is the angular velocity (corr. to $\Delta\boldsymbol{\theta}$).

The complete set of membrane interpolation functions may be expressed as follows:

$$
\begin{aligned}
N^u_{u_a}(\xi,\eta) &= N_a(\xi,\eta) + \sum_{a=1}^{4} \overline{c}^u_{u_a} \overline{N}^u_a(\xi,\eta) \\
N^u_{v_a}(\xi,\eta) &= \sum_{a=1}^{4} \overline{c}^u_{v_a} \overline{N}^u_a(\xi,\eta) \\
N^u_{\theta_{za}}(\xi,\eta) &= \sum_{a=1}^{4} \overline{c}^u_{\theta_{za}} \overline{N}^u_a(\xi,\eta) \\
N^v_{u_a}(\xi,\eta) &= \sum_{a=1}^{4} \overline{c}^v_{u_a} \overline{N}^v_a(\xi,\eta) \\
N^v_{v_a}(\xi,\eta) &= N_a(\xi,\eta) + \sum_{a=1}^{4} \overline{c}^v_{v_a} \overline{N}^v_a(\xi,\eta) \\
N^v_{\theta_{za}}(\xi,\eta) &= \sum_{a=1}^{4} \overline{c}^v_{\theta_{z\varphi}} \overline{N}^v_a(\xi,\eta)
\end{aligned}
\tag{60}
$$

where the position of these functions in the membrane interpolation matrix, $\mathbf{N}^M$, was defined in equation (52).

The above set of 24 coefficients, $\overline{c}^u_{ia}$ and $\overline{c}^v_{ia}$ $(i = 1:3, a = 1:4)$, are computed by evaluating the 8 drilling equations (57)-(58) for each of the following 3 sets of interpolation conditions:

$$
\begin{aligned}
&1) \quad \overline{u}(\xi_a,\eta_a) = \overline{u}_a, \qquad \overline{v}(\xi_a,\eta_a) = 0, \qquad \theta_z(\xi_a,\eta_a) = 0 \\
&2) \quad \overline{u}(\xi_a,\eta_a) = 0, \qquad \overline{v}(\xi_a,\eta_a) = \overline{v}_a, \qquad \theta_z(\xi_a,\eta_a) = 0 \\
&3) \quad \overline{u}(\xi_a,\eta_a) = 0, \qquad \overline{v}(\xi_a,\eta_a) = 0, \qquad \theta_z(\xi_a,\eta_a) = \theta_{za}
\end{aligned}
\tag{61}
$$

where $\overline{u}_a = 1$, $\overline{v}_a = 1$, $\overline{w}_a = 1$ for each node a in turn. This amounts to solving 3 independent 8×8 equation systems – each corresponding to four pairs of interpolation functions: N^u_{ia}, N^v_{ia}, with i fixed at 1,2 or 3 denoting $\overline{u}, \overline{v}$ or $\overline{w}$, respectively. As in the case of the bending interpolation functions, this solution for the membrane interpolation functions is performed only once per element at the beginning of the analysis.

As an aside, note that the bilinear Lagrange interpolation functions, $N_a(\xi,\eta)$, already

satisfy these conditions, *i.e.*,

$$\overline{u}(\xi_a,\eta_a) = \sum_{b=1}^{4} N_b(\xi_a,\eta_a)\overline{u}_b = \delta_{ab}\overline{u}_b = \overline{u}_a$$

$$\overline{v}(\xi_a,\eta_a) = \sum_{b=1}^{4} N_b(\xi_a,\eta_a)\overline{v}_b = \delta_{ab}\overline{v}_b = \overline{v}_a \tag{62}$$

2.6.3 Combined Displacement Field

The complete displacement field for the E410 element may be assembled from the membrane and bending displacement fields, equations (52) and (44) respectively, as follows:

$$\underbrace{\begin{Bmatrix} \overline{u}(\xi,\eta) \\ \overline{v}(\xi,\eta) \\ \overline{w}(\xi,\eta) \end{Bmatrix}}_{\overline{\mathbf{u}}(\xi,\eta)} = \sum_{a=1}^{4} \underbrace{\begin{array}{c} \begin{array}{cccccc} \overline{u}_a & \overline{v}_a & \overline{w}_a & \theta_{xa} & \theta_{ya} & \theta_{za} \end{array} \\ \begin{pmatrix} N^u_{u_a} & N^u_{v_a} & 0 & 0 & 0 & N^u_{\theta_{za}} \\ N^v_{u_a} & N^v_{v_a} & 0 & 0 & 0 & N^v_{\theta_{za}} \\ 0 & 0 & N^w_{w_a} & N^w_{\theta_{xa}} & N^w_{\theta_{ya}} & 0 \end{pmatrix} \end{array}}_{\mathbf{N}^D_a(\xi,\eta)} \underbrace{\begin{Bmatrix} \overline{u}_a \\ \overline{v}_a \\ \overline{w}_a \\ \theta_{xa} \\ \theta_{ya} \\ \theta_{za} \end{Bmatrix}}_{\mathbf{d}_a} \tag{63}$$

where the N^w_{ia} (bending) interpolation functions were defined in equation (45), and the N^u_{ia} and N^v_{ia} (membrane) interpolation functions were defined in equation (60).

Note that an expanded matrix, $\widetilde{\mathbf{N}}^D$, is needed to interpolate the expanded displacement vector, $\widetilde{\mathbf{u}}(\xi,\eta)$, which appears in equations (10) and (11a). The expanded interpolation matrix is defined as follows:

$$\underbrace{\begin{Bmatrix} \overline{u}(\xi,\eta) \\ \overline{v}(\xi,\eta) \\ \overline{w}(\xi,\eta) \\ -\overline{w}_{,x}(\xi,\eta) \\ -\overline{w}_{,y}(\xi,\eta) \end{Bmatrix}}_{\widetilde{\mathbf{u}}(\xi,\eta)} = \sum_{a=1}^{4} \underbrace{\begin{array}{c} \begin{array}{cccccc} \overline{u}_a & \overline{v}_a & \overline{w}_a & \theta_{xa} & \theta_{ya} & \theta_{za} \end{array} \\ \begin{pmatrix} N^u_{u_a} & N^u_{v_a} & 0 & 0 & 0 & N^u_{\theta_{za}} \\ N^v_{u_a} & N^v_{v_a} & 0 & 0 & 0 & N^v_{\theta_{za}} \\ 0 & 0 & N^w_{w_a} & N^w_{\theta_{xa}} & N^w_{\theta_{ya}} & 0 \\ 0 & 0 & -N^w_{w_a,x} & -N^w_{\theta_{xa},x} & -N^w_{\theta_{ya},x} & 0 \\ 0 & 0 & -N^w_{w_a,y} & -N^w_{\theta_{xa},y} & -N^w_{\theta_{ya},y} & 0 \end{pmatrix} \end{array}}_{\widetilde{\mathbf{N}}^D_a(\xi,\eta)} \underbrace{\begin{Bmatrix} \overline{u}_a \\ \overline{v}_a \\ \overline{w}_a \\ \theta_{xa} \\ \theta_{ya} \\ \theta_{za} \end{Bmatrix}}_{\mathbf{d}_a} \tag{64}$$

The above matrix is used to construct the element external force vector (see Section 2.9.2).

2.7 Strain Approximations

The E410 element strains are obtained at the standard 2×2 Gauss integration points by substituting the displacement approximations given above into the shell strain-displacement relations (eqs. (19)-(20)). This requires differentiation of equation (63) and leads to the conventional strain-displacement matrix, $\mathbf{B}$, introduced in equation (25).

As the membrane and bending displacement fields are separable (for geometrically linear analysis), *i.e.*, employing different interpolation functions and involving different nodal degrees of freedom, we may consider the corresponding partitions of the $\mathbf{B}$ matrix separately. We shall denote the *compressed* partitions of $\mathbf{B}$ with a superposed hat (*e.g.*, $\widehat{\mathbf{B}}^{\overline{\epsilon}}$ and $\widehat{\mathbf{B}}^{\kappa}$) and the expanded partitions – corresponding to the full set of element nodal displacement freedoms – without the hat.

2.7.1 Membrane Strains

The membrane strain approximation is represented as follows:

$$\overline{\boldsymbol{\epsilon}}(\xi,\eta) = \left\{ \begin{array}{c} \overline{\epsilon}_x(\xi,\eta) \\ \overline{\epsilon}_y(\xi,\eta) \\ \overline{\epsilon}_{xy}(\xi,\eta) \end{array} \right\} = \left\{ \begin{array}{c} \overline{u}_{,x}(\xi,\eta) \\ \overline{v}_{,y}(\xi,\eta) \\ \overline{u}_{,y}(\xi,\eta) + \overline{v}_{,x}(\xi,\eta) \end{array} \right\} = \widehat{\mathbf{B}}^{\overline{\epsilon}}(\xi,\eta)\,\mathbf{d}^M \tag{65}$$

where $\widehat{\mathbf{B}}^{\overline{\epsilon}}$ is the membrane strain-displacement matrix, obtained by substituting equation (48) into equation (19). In terms of nodal blocks, $\widehat{\mathbf{B}}^{\overline{\epsilon}}$ is defined as follows:

$$\widehat{\mathbf{B}}^{\overline{\epsilon}} = \left[\widehat{\mathbf{B}}_1^{\overline{\epsilon}} \quad \widehat{\mathbf{B}}_2^{\overline{\epsilon}} \quad \widehat{\mathbf{B}}_3^{\overline{\epsilon}} \quad \widehat{\mathbf{B}}_4^{\overline{\epsilon}} \right] \tag{66}$$

where for a given node a:

$$\widehat{\mathbf{B}}_a^{\overline{\epsilon}}(\xi,\eta) = \begin{array}{c} \\ \overline{\epsilon}_x \\ \overline{\epsilon}_y \\ \overline{\epsilon}_{xy} \end{array} \begin{array}{c} \begin{array}{ccc} \overline{u}_a & \overline{v}_a & \theta_{za} \end{array} \\ \left(\begin{array}{ccc} N^u_{u_a,x} & N^u_{v_a,x} & N^u_{\theta_{za},x} \\ N^v_{u_a,y} & N^v_{v_a,y} & N^v_{\theta_{za},x} \\ N^u_{u_a,y} + N^v_{u_a,x} & N^u_{v_a,y} + N^v_{v_a,x} & N^u_{\theta_{za},y} + N^v_{\theta_{za},x} \end{array} \right) \end{array} \tag{67}$$

Note that since the membrane displacement interpolation functions depend explicitly on ξ,η, the Jacobian matrix must be used to obtain the Cartesian x,y derivatives appearing

in equation (67). This transformation is performed as follows:

$$\begin{Bmatrix} \dfrac{\partial N_{ia}^{u_j}(\xi,\eta)}{\partial x} \\ \dfrac{\partial N_{ia}^{u_j}(\xi,\eta)}{\partial y} \end{Bmatrix} = \underbrace{\begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} \end{bmatrix}^{-T}}_{\mathbf{J}} \begin{Bmatrix} \dfrac{\partial N_{ia}^{u_j}(\xi,\eta)}{\partial \xi} \\ \dfrac{\partial N_{ia}^{u_j}(\xi,\eta)}{\partial \eta} \end{Bmatrix} \tag{68}$$

where $N_{ia}^{u_j}$ represents any of the membrane displacement interpolation functions in equation (60), and $\mathbf{J}$ is the Jacobian matrix, computed as follows:

$$\mathbf{J} = \begin{bmatrix} x_1 & \cdots & x_4 \\ y_1 & \cdots & y_4 \end{bmatrix} \begin{bmatrix} \dfrac{\partial N_1(\xi,\eta)}{\partial \xi} & \dfrac{\partial N_1(\xi,\eta)}{\partial \eta} \\ \vdots & \vdots \\ \dfrac{\partial N_4(\xi,\eta)}{\partial \xi} & \dfrac{\partial N_4(\xi,\eta)}{\partial \eta} \end{bmatrix} \tag{69}$$

where the N_a's are the standard bilinear Lagrange interpolation functions, which are used to interpolate the element geometry (see Section 2.5).

2.7.2 Bending Strains

The bending strain approximation may be represented as follows:

$$\boldsymbol{\kappa}(\xi,\eta) = \begin{Bmatrix} \kappa_x(\xi,\eta) \\ \kappa_y(\xi,\eta) \\ \kappa_{xy}(\xi,\eta) \end{Bmatrix} = \begin{Bmatrix} -\overline{w}_{,xx}(\xi,\eta) \\ -\overline{w}_{,yy}(\xi,\eta) \\ -2\overline{w}_{,xy}(\xi,\eta) \end{Bmatrix} = \widehat{\mathbf{B}}^{\boldsymbol{\kappa}}(\xi,\eta)\,\mathbf{d}^B \tag{70}$$

where $\widehat{\mathbf{B}}^{\boldsymbol{\kappa}}$ is the bending strain-displacement matrix, obtained by substituting equation (40) into equation (19). In terms of nodal blocks, $\widehat{\mathbf{B}}^{\boldsymbol{\kappa}}$ is defined as follows:

$$\widehat{\mathbf{B}}^{\boldsymbol{\kappa}} = [\,\widehat{\mathbf{B}}_1^{\boldsymbol{\kappa}} \quad \widehat{\mathbf{B}}_2^{\boldsymbol{\kappa}} \quad \widehat{\mathbf{B}}_3^{\boldsymbol{\kappa}} \quad \widehat{\mathbf{B}}_4^{\boldsymbol{\kappa}}\,] \tag{71}$$

where for a given node a:

$$\widehat{\mathbf{B}}_a^B(\xi,\eta) = \begin{array}{c} \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{array} \begin{array}{c} \begin{array}{ccc} \overline{w}_a & \theta_{xa} & \theta_{ya} \end{array} \\ \begin{pmatrix} -N^w_{w_a,xx} & -N^w_{\theta_{xa},xx} & -N^w_{\theta_{ya},xx} \\ -N^w_{w_a,yy} & -N^w_{\theta_{xa},yy} & -N^w_{\theta_{ya},yy} \\ -2N^w_{w_a,xy} & -2N^w_{\theta_{xa},xy} & -2N^w_{\theta_{ya},xy} \end{pmatrix} \end{array} \tag{72}$$

Recall from Section 2.6.1 that there are two ways of computing the bending interpolation functions, N_a^w – either as a function of x, y (using the solution of a 12×12 equation system) or explicitly as a function of ξ, η. The latter approach is used only when the equation system for the former approach becomes singular – *e.g.*, when two element edges intersect at 45 degrees. When the bending displacement interpolation functions are generated as a function of x, y (approach 1), the second derivatives appearing in equation (72) are immediately available by differentiation. However, when the bending displacement interpolation functions are generated as a function of ξ, η (approach 2), the chain rule must be used to convert second derivatives with respect to ξ, η to corresponding second derivatives with respect to x, y. This chain-rule transformation may be expressed in matrix form as follows:

$$\boxed{\left[\frac{\partial^2 N(x,y)}{\partial \mathbf{x}^2}\right] = \mathbf{J}^{-T} \underbrace{\left[\left[\frac{\partial^2 N(\xi,\eta)}{\partial \boldsymbol{\xi}^2}\right] - \sum_{k=1}^{2} b_k \mathbf{D}_k\right]}_{\left[\frac{\partial^2 N(\xi,\eta)}{\partial \boldsymbol{\xi}^2}\right]'} \mathbf{J}^{-1}} \tag{73}$$

where N represents any of the interpolation functions, N_{ia}^w, in equation (72), and its second-derivative matrices are given in the Cartesian and natural systems as:

$$\left[\frac{\partial^2 N}{\partial \mathbf{x}^2}\right] = \begin{bmatrix} \frac{\partial^2 N}{\partial x \partial x} & \frac{\partial^2 N}{\partial x \partial y} \\ \frac{\partial^2 N}{\partial y \partial x} & \frac{\partial^2 N}{\partial y \partial y} \end{bmatrix} \quad \text{and} \quad \left[\frac{\partial^2 N}{\partial \boldsymbol{\xi}^2}\right] = \begin{bmatrix} \frac{\partial^2 N}{\partial \xi \partial \xi} & \frac{\partial^2 N}{\partial \xi \partial \eta} \\ \frac{\partial^2 N}{\partial \eta \partial \xi} & \frac{\partial^2 N}{\partial \eta \partial \eta} \end{bmatrix} \tag{74}$$

in which $\mathbf{x}$ represents $\{x, y\}$ and $\boldsymbol{\xi}$ represents $\{\xi, \eta\}$. The vector $\mathbf{b}$ is just the Cartesian gradient of N, *i.e.*,

$$\mathbf{b} = \begin{Bmatrix} b_1 \\ b_2 \end{Bmatrix} = \begin{Bmatrix} \frac{\partial N}{\partial x} \\ \frac{\partial N}{\partial y} \end{Bmatrix} = \mathbf{J}^{-T} \begin{Bmatrix} \frac{\partial N}{\partial \xi} \\ \frac{\partial N}{\partial \eta} \end{Bmatrix} \tag{75}$$

where the Jacobian matrix, $\mathbf{J}$, was defined in equation (69). Finally, the matrix $\mathbf{D}_k$ is defined as follows:

$$\mathbf{D}_k = \begin{bmatrix} \frac{\partial^2 x_k}{\partial \xi \partial \xi} & \frac{\partial^2 x_k}{\partial \xi \partial \eta} \\ \frac{\partial^2 x_k}{\partial \eta \partial \xi} & \frac{\partial^2 x_k}{\partial \eta \partial \eta} \end{bmatrix} \tag{76}$$

where the partials in equation (76) can be computed by differentiating equation (35). Equation (73) is derived by differentiating the Jacobian transformation (eq. (68)) and using the chain rule. It is interesting that an incorrect version of this transformation was employed by both Melosh (ref. 3) and Zienkiewicz (ref. 6). They interpreted $\frac{\partial^2 N}{\partial \boldsymbol{\xi}^2}(\xi, \eta)$ as a second rank tensor, and hence used the standard tensor transformation:

$$\frac{\partial^2 N}{\partial \mathbf{x}^2}(x,y) = \mathbf{J}^{-T} \frac{\partial^2 N}{\partial \boldsymbol{\xi}^2}(\xi,\eta)\, \mathbf{J}^{-1} \tag{77}$$

However, $\frac{\partial^2 N}{\partial \xi^2}(\xi,\eta)$ is *not* a second rank tensor – only the *Cartesian* version, $\frac{\partial^2 N}{\partial x^2}(x,y)$, is. Hence, they omitted the term involving $\mathbf{D}_k$ in equation (73), which caused the element to lock and, in turn, to be abandoned by these authors. The replacement of $\frac{\partial^2 N}{\partial \xi^2}$ in equation (77) by $\frac{\partial^2 N'}{\partial \xi^2}$, as in equation (73), eliminates such difficulties.*

In processor ES5, an attempt is always made first to solve for the x, y bending interpolation functions (eq. (45)). However, if the corresponding 12×12 equation system is singular, a switch to the alternate ξ, η formulation is employed automatically – using equation (73) and the explicit ξ, η interpolation functions.

2.7.3 Combined Strain-Displacement (B) Matrix

The complete $\mathbf{B}$ matrix is constructed by assembling the components of $\mathbf{B}^{\overline{\epsilon}}$ and $\mathbf{B}^{\kappa}$ into the appropriate slots, as follows:

$$\widetilde{\epsilon}(\xi,\eta) \;=\; \left\{ \begin{array}{c} \overline{\epsilon}(\xi,\eta) \\ \\ \kappa(\xi,\eta) \end{array} \right\} \;=\; \mathbf{B}(\xi,\eta)\,\mathbf{d}^e \tag{78}$$

where in terms of nodal blocks:

$$\mathbf{B} \;=\; [\mathbf{B}_1 \quad \mathbf{B}_2 \quad \mathbf{B}_3 \quad \mathbf{B}_4] \tag{79}$$

and each nodal block is defined as follows:

---

* Numerical experiments indicate that the x, y bending interpolation functions are well behaved and exhibit little distortion sensitivity until the element edges are within $45° \pm 1°$, at which point the singularity is detected. Then, with the alternate ξ, η interpolation functions, a slight degradation of accuracy is noted at 45°. Note that we do not use the explicit ξ, η formulation outside of this singular range, since these functions mildly violate the constant-curvature patch test for elements that deviate from a strict parallelogram shape – which explains the slight accuracy degradation just noted. However, in the original ξ, η formulation by Melosh (ref. 3), *rigid-body invariance* was also violated, which caused a severe (*i.e.*, "locking") type of degradation.

$$\mathbf{B}_a(\xi,\eta) =$$

$$\begin{array}{c} \\ \overline{\epsilon}_x \\ \overline{\epsilon}_y \\ \overline{\epsilon}_{xy} \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{array} \begin{array}{c} \begin{array}{cccccc} \overline{u}_a & \overline{v}_a & \overline{w}_a & \theta_{xa} & \theta_{ya} & \theta_{za} \end{array} \\ \left(\begin{array}{cccccc} N^u_{u_a,x} & N^u_{v_a,x} & 0 & 0 & 0 & N^u_{\theta_{za},x} \\ N^v_{u_a,y} & N^v_{v_a,y} & 0 & 0 & 0 & N^v_{\theta_{za},y} \\ N^u_{u_a,y} + N^v_{u_a,x} & N^u_{v_a,y} + N^v_{v_a,x} & 0 & 0 & 0 & N^u_{\theta_{za},y} + N^v_{\theta_{za},x} \\ 0 & 0 & -N^w_{w_a,xx} & -N^w_{\theta_{xa},xx} & -N^w_{\theta_{ya},xx} & 0 \\ 0 & 0 & -N^w_{w_a,yy} & -N^w_{\theta_{xa},yy} & -N^w_{\theta_{ya},yy} & 0 \\ 0 & 0 & -2N^w_{w_a,xy} & -2N^w_{\theta_{xa},xy} & -2N^w_{\theta_{ya},xy} & 0 \end{array} \right) \end{array} \tag{80}$$

It can be seen from the above derivations that the bending strains vary at most linearly along any direction within the element, and the membrane strains have a more complicated variation within the element, as shown in Table 5.

Table 5. Strain Variation within the E410 Element

Strain		*Element Type*
Component		**E410**
Memb.	$\overline{\epsilon}_x$	$p_0(\xi) \times p_2(\eta)$
	$\overline{\epsilon}_y$	$p_2(\xi) \times p_0(\eta)$
	$\overline{\epsilon}_{xy}$	$p_2(\xi,\eta)$
Bend.	κ_x	$p_1(x,y)$ (or $p_1(\xi,\eta)^*$)
	κ_y	$p_1(x,y)$ (or $p_1(\xi,\eta)^*$)
	κ_{xy}	$p_1(x,y)$ (or $p_1(\xi,\eta)^*$)

In Table 5, p_i denotes a polynomial of degree i in the specified coordinate x,y or ξ,η; and the asterisk denotes the alternate ξ,η option which is used for the bending strain field only when the coefficient matrix for the constants, c^{kl}_{ia}, in equation (45) is singular – *i.e.*, when the element shape is distorted by approximately 45 degrees at any node.

2.8 Stress Approximations

Since the E410 element within processor ES5 assumes displacements and strains, stresses are computed directly in terms of strains using the constitutive relations. For linear analysis, this amounts to:

$$\tilde{\sigma}(\xi,\eta) \;=\; \tilde{\mathbf{C}}(\xi,\eta)\,\tilde{\epsilon}(\xi,\eta) \tag{81}$$

where (ξ,η) represents an arbitrary element integration point.

2.9 Force Vectors

All element force vectors are constructed using 2×2 Gauss integration.

2.9.1 Internal Force Vector

The element internal force vector is defined as:

$$\begin{aligned} \mathbf{f}_e^{int} &= \int_S \mathbf{B}^T \tilde{\sigma}\, dS \\ &\approx \sum_{g=1}^{4} w_g\, J(\xi_g,\eta_g) \left(\mathbf{B}^T(\xi_g,\eta_g)\,\tilde{\sigma}(\xi_g,\eta_g)\right) \end{aligned} \tag{82}$$

Note that for linear analysis, the above definition is equivalent to $\mathbf{K}_e^{matl}\,\mathbf{d}_e$.

2.9.2 External Force Vector

$$\mathbf{f}_e^{ext} \;=\; \mathbf{f}_e^{body} \;+\; \mathbf{f}_e^{surf} \;+\; \mathbf{f}_e^{line} \tag{83}$$

where the element *body force* vector is defined as:

$$\begin{aligned} \mathbf{f}_e^{body} &= \int_S \tilde{\mathbf{N}}_D^T \tilde{\mathbf{f}}^b\, dS \\ &\approx \sum_{g=1}^{4} w_g\, J(\xi_g,\eta_g) \left(\tilde{\mathbf{N}}_D^T(\xi_g,\eta_g)\,\tilde{\mathbf{f}}^b(\xi_g,\eta_g)\right) \end{aligned} \tag{84}$$

the element *surface force* vector is defined as:

$$\begin{aligned} \mathbf{f}_e^{surf} &= \int_S \tilde{\mathbf{N}}_D^T \tilde{\mathbf{f}}^s\, dS \\ &\approx \sum_{g=1}^{4} w_g\, J(\xi_g,\eta_g) \left(\tilde{\mathbf{N}}_D^T(\xi_g,\eta_g)\,\tilde{\mathbf{f}}^s(\xi_g,\eta_g)\right) \end{aligned} \tag{85}$$

and the element *line force* vector is defined as:

$$\begin{aligned} \mathbf{f}_e^{line} &= \int_L \widetilde{\mathbf{N}}_D^T \widetilde{\mathbf{f}}^\ell \, dL \\ &\approx \sum_{g=1}^{2} w_g \, J1(\xi_g) \left(\widetilde{\mathbf{N}}_D^T(\xi_g) \widetilde{\mathbf{f}}^\ell(\xi_g) \right) \end{aligned} \tag{86}$$

2.10 Stiffness Matrices

The tangent stiffness matrix, defined as $\mathbf{K} = \frac{\partial \mathbf{f}}{\partial \mathbf{d}}$, is the sum of three contributions, *i.e.*,

$$\mathbf{K} = \mathbf{K}^{matl} + \mathbf{K}^{geom} + \mathbf{K}^{load} \tag{87}$$

which are described in the following sections. All contributions are obtained using 2×2 Gauss numerical integration.

2.10.1 Material Stiffness Matrix

The element material stiffness matrix is defined as:

$$\begin{aligned} \mathbf{K}_e^{matl} &= \int_S \mathbf{B}^T \widetilde{\mathbf{C}} \, \mathbf{B} \, dS \\ &\approx \sum_{g=1}^{4} w_g \, J(\xi_g, \eta_g) \Big(\mathbf{B}^T(\xi_g, \eta_g) \, \widetilde{\mathbf{C}}(\xi_g, \eta_g) \, \mathbf{B}(\xi_g, \eta_g) \Big) \end{aligned} \tag{88}$$

where J is the reference surface Jacobian determinant, and all other quantities have been defined in previous sections.

2.10.2 Geometric Stiffness Matrix

The element geometric stiffness matrix is approximated by:

$$\begin{aligned} \mathbf{K}^{geom} \approx \sum_{g=1}^{4} w_g \, J \big[& n_x (\mathbf{g}^{u,x\,T} \mathbf{g}^{u,x} + \mathbf{g}^{v,x\,T} \mathbf{g}^{v,x} + \mathbf{g}^{w,x\,T} \mathbf{g}^{w,x}) + \\ & n_y (\mathbf{g}^{u,y\,T} \mathbf{g}^{u,y} + \mathbf{g}^{v,y\,T} \mathbf{g}^{v,y} + \mathbf{g}^{w,y\,T} \mathbf{g}^{w,y}) + \\ & n_{xy} (\mathbf{g}^{u,x\,T} \mathbf{g}^{u,y} + \mathbf{g}^{v,x\,T} \mathbf{g}^{v,y} + \mathbf{g}^{w,x\,T} \mathbf{g}^{w,y}) + \\ & n_{xy} (\mathbf{g}^{u,y\,T} \mathbf{g}^{u,x} + \mathbf{g}^{v,y\,T} \mathbf{g}^{v,x} + \mathbf{g}^{w,y\,T} \mathbf{g}^{w,x}) \big] \end{aligned} \tag{89}$$

where n_x, n_y, n_{xy} are membrane stress resultants defined in equation (21). The gradient-displacement *row-matrix* $\mathbf{g}^{u_i,x_j}$ interpolates the displacement derivatives from the element

nodal displacement freedoms, *i.e.*,

$$\frac{\partial u_i}{\partial x_j} = \mathbf{g}^{u_{i,x_j}}\,\mathbf{d}^e = \sum_{a=1}^{4} \mathbf{g}_a^{j_{i,x_j}}\,\mathbf{d}_a^e \tag{90}$$

where $\mathbf{d}_a^e$ is the displacement freedom vector at element node a, expressed in the element Cartesian basis. For example, $\mathbf{g}_a^{u_{,x}}$ in terms of interpolation functions defined earlier takes the form:

$$\mathbf{g}_a^{u_{,x}} = \begin{matrix} \overline{u}_a & \overline{v}_a & \overline{w}_a & \theta_{xa} & \theta_{ya} & \theta_{za} \end{matrix} \\ \begin{pmatrix} N^u_{u_a,x} & N^u_{v_a} 0 & 0 & 0 & 0 & N^u_{\theta_{za},x} \end{pmatrix} \tag{91}$$

The other gradient-displacement matrices have a similar form. Derivatives with respect to x and y coordinates for the membrane interpolation functions are treated as in the computation of strain, requiring the inverse Jacobian (see equation (68)).

2.10.3 Load Stiffness Matrix

The load stiffness matrix is defined as $\mathbf{K}^{load} = \frac{\partial \mathbf{f}^{ext}}{\partial \mathbf{d}}$. While this matrix has been implemented in processor ES5, the corresponding theoretical documentation is in preparation.

2.11 Mass Matrices

2.11.1 Consistent Mass Matrix

The element consistent mass matrix is defined as:

$$\begin{aligned} \mathbf{M}_e &= \int_S \widetilde{\mathbf{N}}_D^T \mathcal{I} \widetilde{\mathbf{N}}_D \, dS \\ &\approx \sum_{g=1}^{4} w_g \, J(\xi_g, \eta_g) \Big(\widetilde{\mathbf{N}}_D^T(\xi_g, \eta_g) \mathcal{I}(\xi_g, \eta_g) \widetilde{\mathbf{N}}_D(\xi_g, \eta_g) \Big) \end{aligned} \tag{92}$$

where $\widetilde{\mathbf{N}}_D$ is the expanded displacement interpolation matrix (eq. (64)), and the integrated density matrix, $\mathcal{I}$ is defined as follows:

$$\mathcal{I} = \int_z \begin{matrix} \\ \overline{u} \\ \overline{v} \\ \overline{w} \\ -\overline{w}_{,x} \\ -\overline{w}_{,y} \end{matrix} \begin{matrix} \begin{matrix} \overline{u} & \overline{v} & \overline{w} & -\overline{w}_{,x} & -\overline{w}_{,y} \end{matrix} \\ \begin{pmatrix} \rho & 0 & 0 & -\rho z & 0 \\ 0 & \rho & 0 & 0 & -\rho z \\ 0 & 0 & \rho & 0 & 0 \\ -\rho z & 0 & 0 & \rho z^2 & 0 \\ 0 & -\rho z & 0 & 0 & \rho z^2 \end{pmatrix} \end{matrix} dz \tag{93}$$

where $\rho(x, y, z)$ is the mass density.

Note: The above consistent mass matrix formulation has not yet been implemented in processor ES5.

2.11.2 Lumped (Diagonal) Mass Matrix

Note: The diagonal mass matrix has not yet been implemented in processor ES5. The formulation is similar to that found in processor ES1 (see reference 7).

2.12 Element Nonlinearity

For nonlinear problems, the discrete system of equations given in equation (30) generalizes to (ignoring structural damping and higher-order inertial effects):

$$\mathbf{M}\,\ddot{\mathbf{d}} + \mathbf{f}^{int}(\mathbf{d}) = \mathbf{f}^{ext}(\mathbf{d},\lambda) \tag{94}$$

where $\mathbf{f}^{int}$ and $\mathbf{f}^{ext}$ are now *nonlinear* vector operators. This equation system is then *linearized*, yielding the following linear equation system to be solved at each iteration of a nonlinear analysis:

$$\mathbf{M}\,\delta\ddot{\mathbf{d}} + \mathbf{K}\,\delta\mathbf{d} = \mathbf{f}^{ext}(\mathbf{d},\lambda) - \mathbf{f}^{int}(\mathbf{d}) \tag{95}$$

where $\mathbf{d}$ is the displacement vector connecting the current (reference) configuration to the initial configuration, $\delta\mathbf{d}$ is the iterative change in the displacement vector (to be computed), and $\mathbf{K}$ is the tangent stiffness matrix at either the current configuration (for True-Newton iteration) or some previous configuration (for Modified-Newton iteration).

The nonlinear ES5 element contributions to $\mathbf{M}$, $\mathbf{K}$, $\mathbf{f}^{ext}$ and $\mathbf{f}^{int}$ have the same form as the linear contributions, with the following exceptions:

1) The stress resultant array, $\widetilde{\sigma}$, which appears explicitly in both $\mathbf{f}^{int}$ and in $\mathbf{K}^{geom}$ is computed using the *Green-Lagrange* strain tensor $\mathbf{E}$ defined as follows:

$$\mathbf{E} = \frac{1}{2}\Big[\frac{\partial \mathbf{u}}{\partial \mathbf{X}} + \Big(\frac{\partial \mathbf{u}}{\partial \mathbf{X}}\Big)^T + \Big(\frac{\partial \mathbf{u}}{\partial \mathbf{X}}\Big)^T \frac{\partial \mathbf{u}}{\partial \mathbf{X}}\Big] \tag{96}$$

where $\mathbf{X}$ are the coordinates in the *undeformed* configuration. To simplify the formulation, the nonlinear terms in $\mathbf{u}$ are added only to the shell *membrane* strains. Thus, in geometrically nonlinear analysis, the linear membrane strains, $\overline{\epsilon}$, defined in previous sections are replaced with the following definition:

$$\overline{\epsilon} \longleftarrow \overline{\epsilon} + \overline{\epsilon}^{NL} \tag{97}$$

where

$$\overline{\epsilon}^{NL} = \frac{1}{2}\Big(\frac{\partial \overline{\mathbf{u}}}{\partial \mathbf{X}}\Big)^T \frac{\partial \overline{\mathbf{u}}}{\partial \mathbf{X}} \tag{98}$$

and $\overline{\mathbf{u}}$ and $\mathbf{X}$ are reference-surface displacements and coordinates, respectively. In vector notation and local (element) Cartesian components, this nonlinear contribution to the shell membrane strains becomes:

$$\overline{\boldsymbol{\epsilon}}^{NL} = \left\{ \begin{array}{c} \overline{\epsilon}_X^{NL} \\ \overline{\epsilon}_Y^{NL} \\ \overline{\epsilon}_{XY}^{NL} \end{array} \right\} = \left\{ \begin{array}{c} \frac{1}{2}(\overline{u}_{,X}^2 + \overline{v}_{,X}^2 + \overline{w}_{,X}^2) \\ \frac{1}{2}(\overline{u}_{,Y}^2 + \overline{v}_{,Y}^2 + \overline{w}_{,Y}^2) \\ (\overline{u}_{,X}\overline{u}_{,Y} + \overline{v}_{,X}\overline{v}_{,Y} + \overline{w}_{,X}\overline{w}_{,Y}) \end{array} \right\} \tag{99}$$

where X, Y denote the *undeformed* coordinates in the initial element Cartesian frame, and $\overline{u}, \overline{v}, \overline{w}$ are the total displacements expressed in this same initial basis. The displacement derivatives are computed from the gradients defined previously in Section 2.10.2:

$$\overline{u}_{i,X_j}(\xi,\eta) = \sum_{a=1}^{4} \mathbf{g}_a^{u_i,X_j}(\xi,\eta) \qquad \left(i,j = 1:2 \Rightarrow \begin{array}{c} u,v \\ X,Y \end{array} \right) \tag{100}$$

Once $\overline{\boldsymbol{\epsilon}}^{NL}$ has been added to the linear strains, as in equation (97), the stress resultants can then be computed from the total strains in the same way as during geometrically linear analysis; *e.g.*, for linear-elastic materials:

$$\widetilde{\boldsymbol{\sigma}} \;=\; \widetilde{\mathbf{C}}\,\widetilde{\boldsymbol{\epsilon}} \tag{101}$$

2) When the strain-displacement relations become nonlinear, the $\mathbf{B}$ matrix appearing in the element internal force and material stiffness arrays, $\mathbf{f}_e^{int}$ and $\mathbf{K}_e^{matl}$, also has to be modified. For nonlinear analysis, the $\mathbf{B}$ matrix represents an *incremental* relation between interior strains and nodal displacements, *i.e.*,

$$\delta\widetilde{\boldsymbol{\epsilon}}(\xi,\eta) \;=\; \mathbf{B}(\xi,\eta;\overline{\mathbf{u}})\,\delta\mathbf{d}^e \tag{102}$$

where

$$\delta\mathbf{d}_a = \left\{ \begin{array}{c} \delta\overline{u}_a \\ \delta\overline{v}_a \\ \delta\overline{w}_a \\ \delta\theta_{Xa} \\ \delta\theta_{Ya} \\ \delta\theta_{Za} \end{array} \right\} \tag{103}$$

are the *nodal displacement increments* at node a, and $\overline{\mathbf{u}}$ is the total displacement vector in the interior of the element. As in the case of total strain calculation, the nonlinear contributions to the incremental strains are added only to the membrane components. Thus we replace the linear membrane strain increments as follows:

$$\delta\overline{\boldsymbol{\epsilon}} \;\longleftarrow\; \delta\overline{\boldsymbol{\epsilon}} \;+\; \delta\overline{\boldsymbol{\epsilon}}^{NL} \tag{104}$$

where

$$\delta\overline{\epsilon}^{NL} = \left\{ \begin{array}{l} \overline{u}_{,X}\delta\overline{u}_{,X} + \overline{v}_{,X}\delta\overline{v}_{,X} + \overline{w}_{,X}\delta\overline{w}_{,X} \\ \\ \overline{u}_{,Y}\delta\overline{u}_{,Y} + \overline{v}_{,Y}\delta\overline{v}_{,Y} + \overline{w}_{,Y}\delta\overline{w}_{,Y} \\ \\ \overline{u}_{,X}\delta\overline{u}_{,Y} + \overline{u}_{,Y}\delta\overline{u}_{,X} + \overline{v}_{,X}\delta\overline{v}_{,Y} \\ + \quad \overline{v}_{,Y}\delta\overline{v}_{,X} + \overline{w}_{,X}\delta\overline{w}_{,Y} + \overline{w}_{,Y}\delta\overline{w}_{,X} \end{array} \right\} \tag{105}$$

Substituting the displacement approximations into the above equation leads to the corresponding nonlinear contribution to the **B** matrix; *i.e.*, we replace:

$$\mathbf{B}^{\overline{\epsilon}} \longleftarrow \mathbf{B}^{\overline{\epsilon}} + \mathbf{B}^{\overline{\epsilon}_{NL}} \tag{106}$$

where

$$\mathbf{B}_a^{\overline{\epsilon}_{NL}}(\xi,\eta) =$$

$$\begin{array}{c} \\ \delta\overline{\epsilon}_X \\ \\ \\ \delta\overline{\epsilon}_Y \\ \\ \\ \delta\overline{\epsilon}_{XY} \\ \\ \\ \\ \end{array} \begin{array}{c} \overline{u}_a \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \left(\begin{array}{c} \overline{u}_{,X}N^u_{u_{a,X}} \\ +\overline{v}_{,X}N^v_{u_{a,X}} \\ \\ \\ \overline{u}_{,Y}N^u_{u_{a,Y}} \\ +\overline{v}_{,Y}N^v_{u_{a,Y}} \\ \\ \\ \overline{u}_{,X}N^u_{u_{a,Y}} \\ +\overline{u}_{,Y}N^u_{u_{a,X}} \\ +\overline{v}_{,X}N^v_{u_{a,Y}} \\ +\overline{v}_{,Y}N^v_{u_{a,X}} \\ \\ \\ \end{array} \right. \begin{array}{c} \overline{v}_a \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \left(\begin{array}{c} \overline{u}_{,X}N^u_{v_{a,X}} \\ +\overline{v}_{,X}N^v_{v_{a,X}} \\ \\ \\ \overline{u}_{,Y}N^u_{v_{a,Y}} \\ +\overline{v}_{,Y}N^v_{v_{a,Y}} \\ \\ \\ \overline{u}_{,X}N^u_{v_{a,Y}} \\ +\overline{u}_{,Y}N^u_{v_{a,X}} \\ +\overline{v}_{,X}N^v_{v_{a,Y}} \\ +\overline{v}_{,Y}N^v_{v_{a,X}} \\ \\ \\ \end{array} \right. \begin{array}{c} \overline{w}_a \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \left(\begin{array}{c} \overline{w}_{,X}N^w_{w_{a,X}} \\ \\ \\ \\ \overline{w}_{,Y}N^w_{w_{a,Y}} \\ \\ \\ \\ \overline{w}_{,X}N^w_{w_{a,Y}} \\ +\overline{w}_{,Y}N^w_{w_{a,X}} \\ \\ \\ \\ \\ \\ \end{array} \right. \begin{array}{c} \theta_{Xa} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \left(\begin{array}{c} \overline{w}_{,X}N^w_{\theta_{Xa,X}} \\ \\ \\ \\ \overline{w}_{,Y}N^w_{\theta_{Xa,Y}} \\ \\ \\ \\ \overline{w}_{,X}N^w_{\theta_{Xa,Y}} \\ +\overline{w}_{,Y}N^w_{\theta_{Xa,X}} \\ \\ \\ \\ \\ \\ \end{array} \right. \begin{array}{c} \theta_{Ya} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \left(\begin{array}{c} \overline{w}_{,X}N^w_{\theta_{Ya,X}} \\ \\ \\ \\ \overline{w}_{,Y}N^w_{\theta_{Ya,Y}} \\ \\ \\ \\ \overline{w}_{,X}N^w_{\theta_{Ya,Y}} \\ +\overline{w}_{,Y}N^w_{\theta_{Ya,X}} \\ \\ \\ \\ \\ \\ \end{array} \right. \begin{array}{c} \theta_{Za} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \left(\begin{array}{c} \overline{u}_{,X}N^u_{\theta_{Za,X}} \\ +\overline{v}_{,X}N^v_{\theta_{Za,X}} \\ \\ \\ \overline{u}_{,Y}N^u_{\theta_{Za,Y}} \\ +\overline{v}_{,Y}N^v_{\theta_{Za,Y}} \\ \\ \\ \overline{u}_{,X}N^u_{\theta_{Za,Y}} \\ +\overline{u}_{,Y}N^u_{\theta_{Za,X}} \\ +\overline{v}_{,X}N^v_{\theta_{Za,Y}} \\ +\overline{v}_{,Y}N^v_{\theta_{Za,X}} \\ \\ \\ \end{array} \right) \tag{107}$$

Note that the nonlinear membrane **B** matrix now couples all of the element freedoms, and also depends on the nodal displacements through the displacement derivatives, which are computed using equation (100).

3) All of the element integrals, *e.g.*, for stiffness matrices and force vectors, are carried out over the *initial* (undeformed) element domain. The effect of large rotations is accounted for using $\overline{\epsilon}^{NL}$ and $\mathbf{B}^{\overline{\epsilon}_{NL}}$, defined above.

4) For very large rotations, the *corotational* facility built-in to the generic element processor shell (ES) may be used. In this case the bulk rigid body motion of each element is first "subtracted" from the overall motion before computing $\mathbf{K}_e$, $\mathbf{f}_e^{int}$, and $\widetilde{\sigma}(\widetilde{\epsilon})$. The main effect of this adjustment is to increase the accuracy of the nonlinear strain-displacement relations (eq. (97)), since the nonlinear terms in the Green-Lagrange strain tensor, *i.e.*, $\overline{\epsilon}^{NL}$, become small after the element's rigid body motion has been subtracted; and the accuracy continues to increase as the

shell element mesh is refined. In fact, with the corotational option on, it is even possible to solve nonlinear problems without using any other element nonlinearity (albeit with a lower order of accuracy). See reference 5 for details.

4) The element external force vector, $\mathbf{f}_e^{ext}$, is a nonlinear function of the displacement vector, $\mathbf{d}_e$, only if *live* (*e.g.*, hydrostatic) loads are present. For displacement-independent (*i.e.*, *dead*) loads, the external force vector is usually expressible as:

$$\mathbf{f}^{ext} = \lambda \widehat{\mathbf{f}}^{ext} \tag{108}$$

where $\widehat{\mathbf{f}}^{ext}$ is a fixed base load vector, and λ is the current load factor. In this case, equation (85) is evaluated only once (initially), and scaled by λ as the analysis progresses.

3. REFERENCES

1. Stewart, Caroline B., Compiler: *The Computational Structural Mechanics Testbed User's Manual.* NASA TM-100644, October, 1989.

2. Almroth, B.O., Brogan, F.A. and Stanley, G.M.: *Structural Analysis of General Shells, Vol. II: User Instructions for the STAGS(C-1) Computer Code*, NASA CR-165671, 1981.

3. Melosh, R.J.: "Basis of Derivation of Matrices for the Direct Stiffness Method," *AIAA Journal* , vol. 1, pp. 1631-1637, 1963.

4. Willam, K.J.: "Finite Element Analysis of Cellular Structures," Ph.D. Thesis, Department of Civil Engineering, U.C., Berkeley, CA 1969.

5. Stanley, Gary and Nour-Omid, Shahram: *The Computational Structural Mechanics Testbed Generic Structural-Element Processor Manual.* NASA CR-181728, 1990.

6. Zienkiewicz, O.C.: *The Finite Element Method*, McGraw-Hill, London, pp. 234-239, 1977.

7. Stanley, Gary: *The Computational Structural Mechanics Testbed Structural Element Processor ES1: Basic SRI and ANS Shell Elements*, NASA CR-4357, 1991.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-4358		

4. Title and Subtitle	5. Report Date
The Computational Structural Mechanics Testbed Structural Element Processor ES5: STAGS Shell Element	May 1991
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Charles Rankin and Frank Brogan	LMSC-D878511
9. Performing Organization Name and Address	10. Work Unit No.
Lockheed Missiles & Space Company, Inc. Research and Development Division 3251 Hanover Street Palo Alto, California 94304	505-63-53-01
	11. Contract or Grant No.
	NAS1-18444
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	Contractor Report
	14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor: Jerrold M. Housner

16. Abstract

The purpose of this manual is to document the theory behind the CSM Testbed structural finite element processor ES5 for the STAGS shell element. This report is intended both for CSM Testbed users, who would like theoretical background on element types before selecting them for an analysis, as well as for element researchers who are attempting to improve existing elements or to develop entirely new formulations.

Processor ES5 contains the displacement-based 4-node quadrilateral shell element used as the "workhorse" element in the STAGS code). This C^1 element is intended for modeling thin shells for which transverse-shear deformation is not important. The element employs an incompatible displacement field, which features a high-order (cubic) bending field and a semi-lower-order (linear/cubic) membrane field. For many problems with regular geometry, the STAGS 410 element in ES5, called E410, provides rapid convergence not possible with the same number of freedoms in an element where inter-element compatibility is rigidly enforced. However, it does exhibit distortion sensitivity, especially due to in-plane distortion. Sensitivity to out-of-plane distortion (*i.e.*, warping) is largely eliminated by the generic corotational projection operator built-in to the ES processor shell.

The E410 is a quadrilateral shell element with 3 translational and 3 rotational freedoms per node. This element also has "drilling" rotational stiffness, which proves to be an advantage for problems with intersecting geometries. Hence, no suppression is required for the drilling (*i.e.*, 6th) degree of freedom – even for flat plates – because all 6 nodal freedoms have stiffness.

17. Key Words (Suggested by Authors(s))	18. Distribution Statement
Structural analysis software Finite Element Implementation Corotational Formulation CSM Testbed System	Unclassified—Unlimited Subject Category 39

19. Security Classif.(of this report)	20. Security Classif.(of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	44	A03

NASA FORM 1626 OCT 86

For sale by the National Technical Information Service, Springfield, Virginia 22161-2171

NASA-Langley, 1991